

Question1:

A university's Office of Admission keeps track of student majors. Each student can have a single major. Below is an example of how their system stores students, majors, and how it manipulates them:

```
students = [("Allen Anderson", "Computer Science"),
("Edgar Einstein", "Engineering"),
("Farrah Finn", "Fine Arts")]
def add_new_student(students, name, major):
    students.append((name, major))
# students[len(students)] = (name, major)
def update_student(students, index, name, major):
    students[index] = name, major
def find_students_by_name(students, name):
    return [student for student in students if name in student[0]]
def get_all_majors(students):
    return [student[1] for student in students]
```

Solution:

What can be concluded from the code snippet above? (Select all acceptable answers.)

1. In the update_student function, the '(' and ')' parentheses can be removed without causing any errors.

--True

2. Calling find_students_by_name(students, 'in') returns a list of 2 tuples.

-- True [('Edgar Einstein', 'Engineering'), ('Farrah Finn', 'Fine Arts')]

3. The add_new_student function can be rewritten as seen below and still maintain identical functionality: students[len(students)] = (name, major)

--False (list assignment index out of range)

4. Calling `get_all_majors(students)` returns a list of 3 tuples.

--False

5. The `add_new_student` function adds a new student in the last place in the list.

--True

6. The name of the first student in the array can be set to the `new_name` variable, like

```
students[0][0] = new_name
```

--False ('tuple' object does not support item assignment)